

## 8

### Considerações Finais

Este trabalho apresenta o problema de aplicações que precisam solicitar ajuste elástico com requisitos bem definidos, como o processamento em tempo de exibição, manipulando fluxos de mídia comprimidos, sem se tornar dependente do decodificador e gerando mídias de alta fidelidade utilizando um fator de ajuste variável. Como foi anteriormente discutido, nenhuma das soluções de ajuste existentes na literatura atendem satisfatoriamente a tais requisitos e isso motiva a definição de novos algoritmos de ajuste voltados para essas aplicações.

Com essa motivação, este trabalho propõe algoritmos para ajuste elástico, em tempo de exibição e compilação, para fluxos de áudio comprimidos. Os algoritmos atuam diretamente sobre fluxos comprimidos sem utilizar mecanismos de decodificação e recodificação e com independência do exibidor de conteúdo. O ajuste elástico é proposto para um modelo genérico de fluxos de áudio e então instanciado para diversos padrões de áudio de alta qualidade atualmente utilizados.

Além desses, também são propostos algoritmos para manter o sincronismo intermídia quando o ajuste é aplicado em fluxos de sistemas. Esse algoritmo permite que as operações de ajuste elástico em cada mídia possam ser realizadas de forma independente, evitando-se o dilema da escolha de um instante específico em que o ajuste deva ocorrer simultaneamente em todos os fluxos individuais.

A ferramenta de ajuste permite que outras aplicações possam facilmente solicitar serviços interagindo apenas com suas APIs de ajuste. Além de implementar os algoritmos propostos, a ferramenta está integrada ao algoritmo desenvolvido por Cavendish (Cavendish, 2005), o que possibilita também oferecer o ajuste elástico em fluxos de vídeo MPEG-2. Vale ressaltar que as APIs de ajuste propostas são genéricas para serem implementadas de diferentes maneiras. A implementação realizada por este trabalho é apenas um exemplo.

Como estudo de caso, este trabalho discute como a ferramenta de ajuste pode ser integrada a sistemas hipermídia e realiza a integração com o sistema HyperProp através da construção de adaptadores de exibidores de conteúdo. A presença da ferramenta de ajuste permite que o formatador possa exibir corretamente hiperdocumentos que, de outro modo, não poderiam ter a consistência temporal preservada.

Algumas medidas de qualidade foram extraídas do algoritmo de ajuste. A elaboração de testes para comparar diferentes áudios com ajuste elástico processados por diferentes algoritmos é muito rara na literatura. Os resultados obtidos mostram que a qualidade do algoritmo proposto se aproxima à do *Sound Forge*, o que é um ótimo resultado, uma vez que o algoritmo proposto funciona em tempo de exibição e pode ser facilmente integrado a uma outra aplicação, ao contrário do *Sound Forge*, e que o *Sound Forge* é uma ferramenta bastante utilizada atualmente, inclusive em aplicações comerciais de processamento de áudio. Vale lembrar que em aplicações para manutenção das relações de sincronismo, o ajuste é aplicado apenas em pequenas partes do fluxo de áudio, o que reduz o efeito da queda de qualidade decorrente do processamento. Além disso, o tempo de processamento para executar o algoritmo proposto é muito menor do que o do *Sound Forge* e a duração atingida é mais próxima da ótima.

Amostras de áudios processados com o algoritmo proposto podem ser encontradas na seguinte URL:

<http://www.telemidia.puc-rio.br/~smbm/ajusteaudio>

A análise da implementação demonstra que a ferramenta de ajuste é extensível, principalmente para novos formatos de áudio. Além disso, aplicações clientes do ajuste elástico devem sempre considerar o retardo inicial que precisa ser introduzido quando é solicitado o processamento de ajuste em tempo de exibição.

Alguns trabalhos futuros precisam ainda ser desenvolvidos. O principal deles é terminar a integração do algoritmo de ajuste para fluxos de sistemas desenvolvido com o algoritmo de ajuste de vídeo de Cavendish (Cavendish, 2005). Isso implica não somente na criação de classes de integração como em algumas modificações no algoritmo de vídeo.

Outro trabalho importante é a criação de novos testes de qualidade subjetiva para medir a qualidade utilizando outros formatos de áudio ou mesmo MP3 com

diferentes taxa de bits e de amostragem. Também é importante definir o real limite de fator de ajuste que o algoritmo utilizar com qualidade aceitável e não simplesmente supor que o valor é 10%. Ademais, uma vez terminada a integração com o algoritmo de vídeo, é possível realizar testes com fluxos audiovisuais.

Além disso, algumas extensões aos algoritmos de ajuste importantes são generalizar o algoritmo de sistemas para manipular o formato do Fluxo de Transporte do MPEG-2 e o algoritmo de áudio para suportar o perfil principal do AAC. O algoritmo de ajuste de áudio também pode suportar novos formatos, como o OGG-Vorbis.

Em relação à implementação, alguns trabalhos futuros são:

- O algoritmo de verificação do sincronismo intermídia incluindo o tratamento de situações de erro, quando a ferramenta de ajuste não consegue atingir o fator desejado, sem perder o sincronismo intermídia;
- A criação ou remoção de estruturas de sistemas, se necessário, durante a realização de ajuste;
- O controle para que o intervalo máximo entre as amostras de relógio transmitidas em fluxos de sistemas não ultrapasse o tempo máximo definido;
- O suporte a fluxos AAC codificados em outros protocolos de transporte e alguns casos bem particulares de fluxos MPEG.

A comunicação com exibidores de conteúdo também precisa ser melhorada. Atualmente, ainda não existe uma comunicação que funcione de modo integrado à aplicação cliente com suporte a vários formatos. Para solucionar esse problema, é necessário desenvolver uma integração entre o VLC e um adaptador *FlexMediaDataSourceAdapter* utilizando Java JNI (*Java Native Interface*).

Por fim, embora a ferramenta de ajuste já esteja sendo utilizada pelo formatador HyperProp, é interessante criar novos exemplos de integração, utilizando as diversas funções da ferramenta de ajuste, principalmente explorando o processamento em tempo de exibição.